

```
1 ////////////////////////////////////////////////////
2 // Компоненты - Поля
3 ////////////////////////////////////////////////////
4 // важные аспекты:
5 // 1. предложения по дополнению или изменениям структуры приветствуются
6 // 2. поля и контейнеры могут использоваться наравне
7 // 3. компоненты полей будут использованы в формах и отдельно от форм, вывод
  значения verbose_name не должен припятствовать этому
8
9
10 //Базовые свойства для каждого поля
11 {
12     "component":"XXXXXXXField", //класс компонента
13     "verbose_name":"Адрес", //Видимое название, поле для пользователя
14     "required":"true", //обязательность заполнения поля, default=false
15     "readonly":"false", //поле в режиме только чтение, default=true
16     "help_text":"Укажите полный адрес доставки", //Тест краткой справки о
    заполнении поля
17
18     "normalizers":[ //список фильтров, приводящих значение в нормальную форму или
    сообщающих об ошибке
19         //примеры:
20         {"normalizer":"lowercase"}, //приводит в нижний регистр
21         {"normalizer":"string_length", "min":5, "max":10}, //проверка размера строки
22         {"normalizer":"email_format"},
23     ]
24 }
25
26
27 //StringField - строковое поле
28 {
29     "component":"StringField", //класс компонента
30     "max_length":255, //в принципе, это поле может просто установить или
    переопределить внутренний - {"normalizer":"string_length"
31     "prefix":""," //префикс перед импутом см. Input group, bootstrap.
32     "sofix":""
33 }
34
35
36 //IntegerField - целочисленное поле (предположительно <input type="number">)
37 {
38     "component":"IntegerField", //класс компонента
39     "min":-100, //необязательное поле, минимальное допустимое значение
40     "max":100, //необязательное поле, максимально допустимое значение
41     "step":1 //шаг изменения числа стрелками вверх/вниз
42 }
43
44
45
46 //FloatField - поля ввода числа с плавающей точкой
47 {
48     "component":"FloatField", //класс компонента
49 }
50
51
52 //TextField - текстовое поле (предположительно <textarea>)
53 {
54     "component":"TextField", //класс компонента
55 }
56
57 //BooleanField - флаг
58 {
59     "component":"BooleanField", //класс компонента
60 }
61
62 //SelectField - выбор из списка <select>
63 {
64     "component":"SelectField", //класс компонента
65     "static_values":{" //статический список значений
66         "val1":"Видимое значение1",
67         "val2":"Видимое значение2",
68     },
69     "dynamic_values":{
```

```

70         //тут нужно согласовать лучший подход получения данных с сервера
71     }
72 }
73
74 ////////////////////////////////////////////////////
75 // Компоненты - контейнеры
76 ////////////////////////////////////////////////////
77
78 {"component":"FieldSetContainer", //Контейнер из полей визуальнo выделенный в рамку
  с названием, <fieldset>
79 "legend":"Блок важных полей",
80 "childs":[
81     //список любых компонентов полей и/или контейнеров
82 ]}
83
84
85 {"component":"TabsContainer", //контейнер из вкладок
86 "tabs":[
87     {"verbose_name":"Вкладка 1",
88      "icon":"glyphicon glyphicon-wrench", //bootstrap иконка или сделанная
      аналогично, необязательное поле
89      "childs":[
90          //список любых компонентов полей и/или контейнеров
91          ]},
92     {"verbose_name":"Вкладка 2",
93      "childs":[
94          //список любых компонентов полей и/или контейнеров
95          ]}
96 ]}
97
98
99 {"component":"IfContainer", //контейнер, содержащий одно поле которое будет
показано при выполнении условия
100 "condition":"field1=value & field2=value3", //условие отображения контейнера,
имена переменных=имена полей значение переменных=значение полей в момент проверки
101 "then":[
102     //список полей и контейнеров которые будут отображены при выполнении условия
103 ]}
104
105
106 ////////////////////////////////////////////////////
107 // Фильтры - нормализаторы =)
108 ////////////////////////////////////////////////////
109 //
110 // Последовательность применения нормализаторов имеет значение
111
112
113 {"normalizer":"lowercase"} //приводит текст в нижний регистр
114 {"normalizer":"uppercase"} //приводит текст в верхний регистр
115 {"normalizer":"capfirst"} //приводит первую букву в верхний регистр
116
117 {"normalizer":"string_length", //проверка размера строки, не соответствие вывод ошибки
118 "min":5,
119 "max":10}
120
121 {"normalizer":"email_format"} //проверка формата email, если не соответствует то
вывод ошибки
122
123
124
125
126
127

```